

### 编译原理实验二

**语义分析**

学 院 计算机科学与技术

学　　 号

班　　 级

学 生

日 期 2022年4月9日

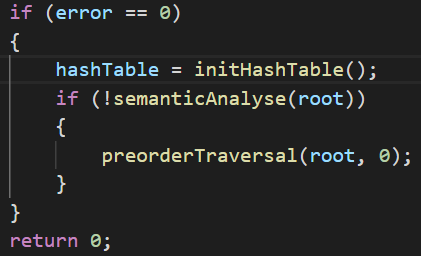
## 编写文件列表

在实验一的基础上添加语义分析函数头文件和文件：

analyse.h：定义语义分析辅助枚举、结构体、数据结构等，声明语义分析函数。

analyse.c：定义语义分析函数。

编写语义分析主函数semanticAnalyse，在没有词法和语法错误时传入词法和语法分析得到的分析树根对分析树进行分析，返回语义错误数；并修改词法分析主程序main.c，令其在不出现语义错误时输出分析树。



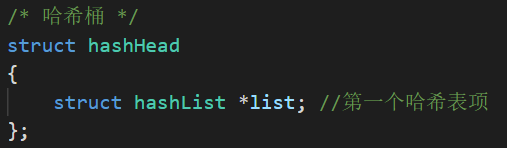
## 程序实现功能

#### 2.1符号表

利用实验指导书给出的哈希函数hash\_pjw，编写哈希表作为符号表。表项结构为符号名和符号类型，以及连接下一个表项的指针。相同哈希值(冲突)的表项放入同一个哈希桶中，使用指针连接。

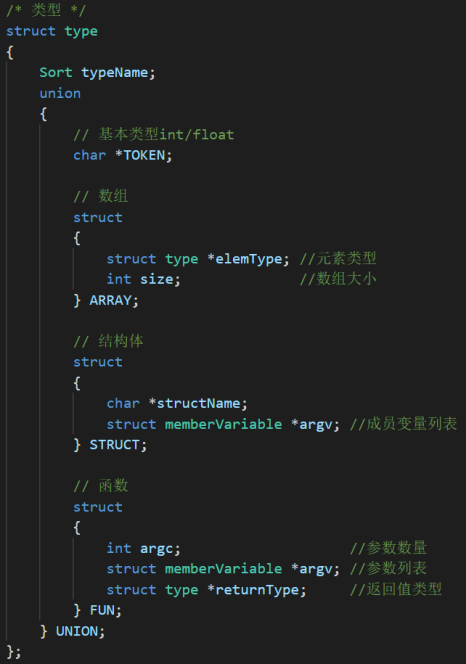
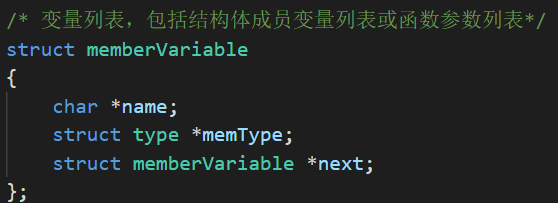


哈希表使用全局变量的哈希桶数组，桶结构有指向第一个哈希表项的指针，数组下标表示该桶内哈希表项的哈希值。全程序只使用一个符号表存储所有符号。

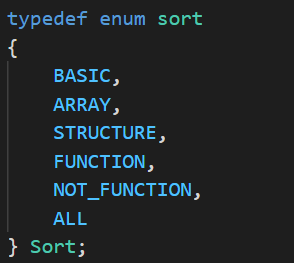
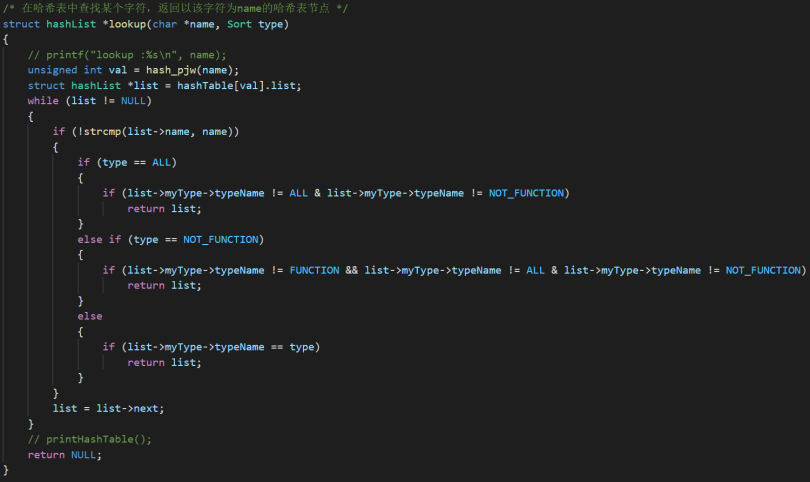


#### 2.2符号表示

符号类型使用结构体type表示，根据类型不同参数不同，使用union节省空间。对结构体、函数等需要保存成员变量、参数列表等符号列表的类型，定义结构体memberVariable使用链表的形式存储变量列表，在类型结构体的argv变量中只存储链表第一个节点。

由于只使用一个符号表存储所有符号，定义sort枚举类型对表内所有符号进行分类，分为基本类型、数组、结构体、函数、非函数、任意类型。防止同名不同类型的符号冲突以及lookup查询同名符号(必冲突)时找到错误类型。

在符号表中使用lookup根据符号名和对应类型名(sort)寻找符号表项，使用哈希函数得到该符号名对应哈希值，并在哈希表与该哈希值对应的桶的表项链表中遍历寻找该节点。

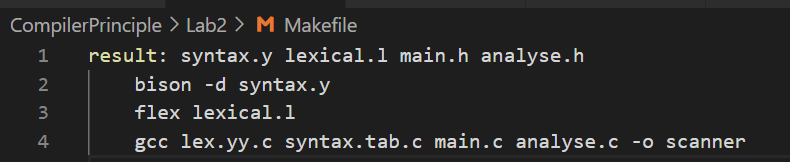
#### 2.3过程函数

根据产生式编写可以进行分析的非终结符对应的过程函数。当在产生式右部遇到可以分析的非终结符时，递归调用该终结符对应的过程函数，对对应节点分析并返回分析结果；当遇到可以直接分析的情况时，迭代对该节点分析。

这里情况过多，重点为根据产生式取出适当的分析树节点，根据节点中保存的成员变量，或构造新的符号表项，或取出对应符号表项进行分析对比。

## 编译和使用方法

使用Makefile进行编译，进入目录后在终端输入make，按照Makefile文件内进行编译和链接后，生成可执行文件scanner。



使用方法：./scanner [待分析文件名]

例： ./scanner input1.cmm

程序可正确分析通过必做的17个测试样例。